BNL-79366-2007-CP

Hard Real Time Quick EXAFS Data Acquisition With All Open Source Software On A Commodity Personal Computer

I. So, D.P. Siddons, W.A. Caliebe and S. Khalid

Proceedings of Synchrotron Radiation Instrumentation (SRI 2007)
Baton Rouge, LA / April 25 - 27, 2007

October 2007

National Synchrotron Light Source

Brookhaven National Laboratory
P.O. Box 5000
Upton, NY 11973-5000
www.bnl.gov

DISCLAIMER

BNL-79366-2007-CP

Hard Real Time Quick EXAFS Data Acquisition With All Open Source Software On A Commodity Personal Computer

I. So, D.P. Siddons, W.A. Caliebe, and S. Khalid

National Synchrotron Light Source, Brookhaven National Laboratory, Upton, NY 11973, USA
Hasylab am Desy, Notkestrasse 85, 22603 Hamburg, Germany

Abstract

We describe here the data acquisition subsystem of the Quick EXAFS (QEXAFS) experiment at the National Synchrotron Light Source of Brookhaven National Laboratory. For ease of future growth and flexibility, almost all software components are open source with very active maintainers. Among them, Linux running on x86 desktop computer, RTAI for real-time response, COMEDI driver for the data acquisition hardware, Qt and PyQt for graphical user interface, PyQwt for plotting, and Python for scripting. The signal (A/D) and energy-reading (IK220 encoder) devices in the PCI computer are also EPICS enabled. The control system scans the monochromator energy through a networked EPICS motor.

With the real-time kernel, the system is capable of deterministic data-sampling period of tens of micro-seconds with typical timing-jitter of several micro-seconds. At the same time, Linux is running in other non-real-time processes handling the user-interface. A modern Qt-based controls-frontend enhances productivity. The fast plotting and zooming of data in time or energy coordinates let the experimenters verify the quality of the data before detailed analysis. Python scripting is built-in for automation. The typical data-rate for continuous runs are around ten mega-bytes per minute.

Introduction

Computer hardware performance per dollar is increasing at an exponential rate. How can we take best advantage of it for scientific data-acquisition (daq) at a multi-beamline facility like the National Synchrotron Light Source (NSLS) ?

For the NSLS-run beamlines, the default computer control system is now the Experimental Physics and Industrial Control System (EPICS) [1]. Other specialized systems (e.g. Windows only detector) are made to inter-operate with EPICS when needed. The current standard beamline EPICS setup includes RTEMS[2] on VME as IOC and EPICS clients on GNU/Linux[3] hosts.

In this paper we describe a new addition to the standard setup--a Personal Computer (x86 PC) based EPICS Linux client and server with hard real time (RT) capabilities through RTAI[4]. The successful application of this daq subsystem to the QEXAFS experiment demonstrates how we may speedup a traditional synchrotron experiment by a hundred time with today's computer technology. Central to the cost-effectiveness and ease of future-growth of this approach is the fact that all software components (operating systems, device drivers, EPICS and Qt[5] GUI toolkits, ...) of the daq system are open-source. Such a system will benefit the most from future computing advances for the purpose of scientific daq.

The Approach

EPICS has proven itself capable of controlling accelerators and scientific experiments. APS[6] and DIAMOND[7] are obvious examples. The RT EPICS servers are usually run separately from the non-RT EPICS clients. Under Linux, it is possible to run both non-RT server and client on the same machine. This paper shows how we have added RT processes to run on the same Linux PC at the same time. The non-RT EPICS Linux server and client can make use of the RT processes for daq.

Sometimes daq-hardware device drivers only exist for Linux, not RTAI. In such a situation, we have modified the non-RT Linux driver to work with RTAI so we may have deterministic data-sampling. Sampling periods in the high tens of micro-seconds with jitter of less than ten micro-seconds can comfortably be achieved while many other non-RT processes, including Qt based GUI, are displaying and interacting with users.

The Software Architecture

Figure 1 shows the composition of the resulting Graphical Realtime EPICS Controller (GREC) constructed with all open-source components running on a x86 PC.

The regular Linux kernel source is patched with a small (~ 200 K bytes) RTAI Hardware Abstraction Layer (HAL) patch so it may run as the idle-task of the RTAI. With this modification ALL Linux software can be used as usual. Only now the Linux applications have gained the option of making use of the real-time services provided by RTAI.

COMEDI is a set of Linux/RTAI device drivers for daq hardware. An easy to use API is exposed through comedilib. It can be used in Linux/RTAI, user/kernel space as needed. Qt is a state-of-the-art multi-platform (Linux, Windows, Mac OS X) GUI toolkit with all source-code included and a GPL licensing option. A Python binding of it (PyQt [8]) is used in our GUI. EPICS enables the GREC to be part of a distributed controls network when necessary. EPICS device-support exists for the GREC hardware so other EPICS clients can make use of them. The GREC is also capable of controlling through EPICS channel-access.

The QEXAFS Experiment DAQ Subsystem

When local bus motor-driver is not ready on the GREC, it tunes the NSLS X18B monochromator [9] through EPICS networked VME-based OMS motor. The APS motor-record with its associated MEDM screens get things started with little effort. That is the advantage of being part of a mature controls framework. We have since replaced the VME and host computer in the experiment [9] with GREC.

For the analog to digital converters (ADC), we use the COMEDI [10] ni_pcimio driver for the National Instrument multi-function daq board. An EPICS Linux server makes the devices available as EPICS channels. For sub-millisecond sampling periods, RTAI tasks are used. The Linux device-driver for the monochromator angular encoder (Heidenhain IK220) readout is modified to use RTAI tasks so that the sampling-period jitter is below 10 micro-seconds. The hard RT sampling is important for data-quality here because, unlike the traditional step-scan EXAFS approach, the energy-scan in QEXAFS never stops for the ADC to acquire the signal from the detectors. Sub-millisecond sampling allows us to take a snapshot of the whole spectrum in seconds with reasonable signal-to-noise-ratio. The beam-current drop and instrument-drifts are small compared to that of the half -hour scan of EXAFS. Time-dependent reactions in the second-scale can be studied with this daq system.

The sampling-period itself can be varied from 200 micro-second up. We typically operate it at 500 micro-seconds with the monochromator-scanning at a fifth to four cycles per second (we use cyc/sec for mono-scan, Hz for data frequency to avoid confusion). Figure 2 shows a zoomed 1.5 second result of a 30 second scan of Nickel-foil EXAFS at about 1/5 cyc/sec. The highest frequency component in the data (around the K-edge) is less than 200 Hz for a 4 cyc/sec scan.

The figure is plotted on GREC with a PyQt based Python program. The PyQwt widget [11] allows us to zoom in and out to any depth into any rectangular region of the plot by selecting it with the mouse. Individual curves could be switched off/on by the buttons at the bottom of the plot. The "Plot Table" control panel lets us either plot against the time or the energy on a linear or logarithmic scale. Simple operation like plotting (Io / It) is possible (figure 2).

Taking data is activated by clicking the "Take Data" button at the top. Acquiring multiple data-set with auto-set-numbering is possible if the "Loop" variable is larger than one. Daq-timing can be controlled with the "Timing" control panel. Python macros can be entered on the "Macro" panel. The data is displayed during the daq on GREC according to the settings of the "Plot Table" panel so that the user may visually verify the quality of it within seconds with the infinite zoom capability. Several user groups have taken many gigabytes of data with the system in March, 2007. The system has been reliable for the week-long daq in catalysis studies.

Future Tasks

So the GREC works. Its interface is kept as simple as possible so the beamline users may start taking useful data within minutes of training. Since all software components are open-source (most are GPLed) and on a PC, users could get a free copy of it to run on their own computer with simulated data-feed if desired. For demanding tasks, the GREC could grow incrementally up to a full blown EPICS control system; enough to control a mile-size accelerator.

GREC is running the EPICS server under Linux now. A tighter EPICS RTAI integration would allow RT EPICS IOC functionalities with powerful (e.g. gigabytes of RAM for 64-bit RT kernel space) hardware. A better EPICS Qt integration (a QDM to rejuvenate MEDM ?) could modernize EPICS GUI for the next decade. An ongoing minor upgrade for the QEXAFS experiment is to use simultaneous sampling A/D with on-board timer and buffer so we may average over multiple A/D data-samples for noise-reduction while waiting for the slow hundred-micro-second monochromator-energy reading.

Since GREC daq and disk-I/O occur on the same machine, the data-rate in the near future can approach the PCI express bus and RAID-controller bandwidth which is a duplexed 2 GB/s .

Another bonus for the GREC approach is powerful 3D accelerated graphics hardware. A quick 3D-plot of time-dependent EXAFS spectra during daq should speed up data-quality verification.

Conclusion

GREC is designed to be the lowest entry-cost, self contained GUI scientific daq system. It allows scientists to have a reliable, deterministic hard real-time daq system with just about ten micro-second worst case timing-jitter or interrupt-latency. The Qt based GUI enhances productivity. The built-in Python macro interpreter encourages automation. GREC could expand into an industrial strength EPICS system if necessary. It is future-proof'ed with actively maintained software components that are going to adapt to the most cost-effect daq hardware.

Acknowledgements

We would like to express our gratitude to K. Feng for her VME support, B. Clay for helping in noise-diagnostics, Z. Yin for setting up GREC for X18B and A. Lenhard for mechanical device support.

References

[1] EPICS--Experimental Physics and Industrial Control System: http://lansce.lanl.gov/EPICSdata/epics.htm
[2] RTEMS--Real-Time Operating System for Multiprocessor Systems: http://www.rtems.com
[3] GNU/Linux: GNU: http://www.gnu.org; Linux: http://www.kernel.org
[4] RTAI—RealTime Application Interface for Linux: https://www.rtai.org
[5] Qt—Trolltech's cross-platform (Windows, Unix/X11, Mac OS X) applications framework http://doc.trolltech.com/4.2 (Open Source Edition)
[6] APS—Advanced Photon Source: http://www.aps.anl.gov/epics
[7] DIAMOND—Diamond Light Source Ltd: http://www.diamond.ac.uk/Technology/Controls/EPICS.htm
[8] PyQt--Python bindings for Trolltech's Qt application framework http://www.riverbankcomputing.co.uk/pyqt/index.php
[9] NSLS X18B QEXAFS Setup: W.A. Caliebe Cam-driven Monochromator for QEXAFS [Radiation Physics and Chemistry Vol 75, 11, Nov 2006, p 1962-1965 Elsevier Science]
[10] COMEDI--open-source drivers, tools, and libraries for data acquisition: http://www.comedi.org
[11] PyQwt--Python bindings for the Qwt C++ class library for scientific and engineering applications http://pyqwt.sourceforge.net

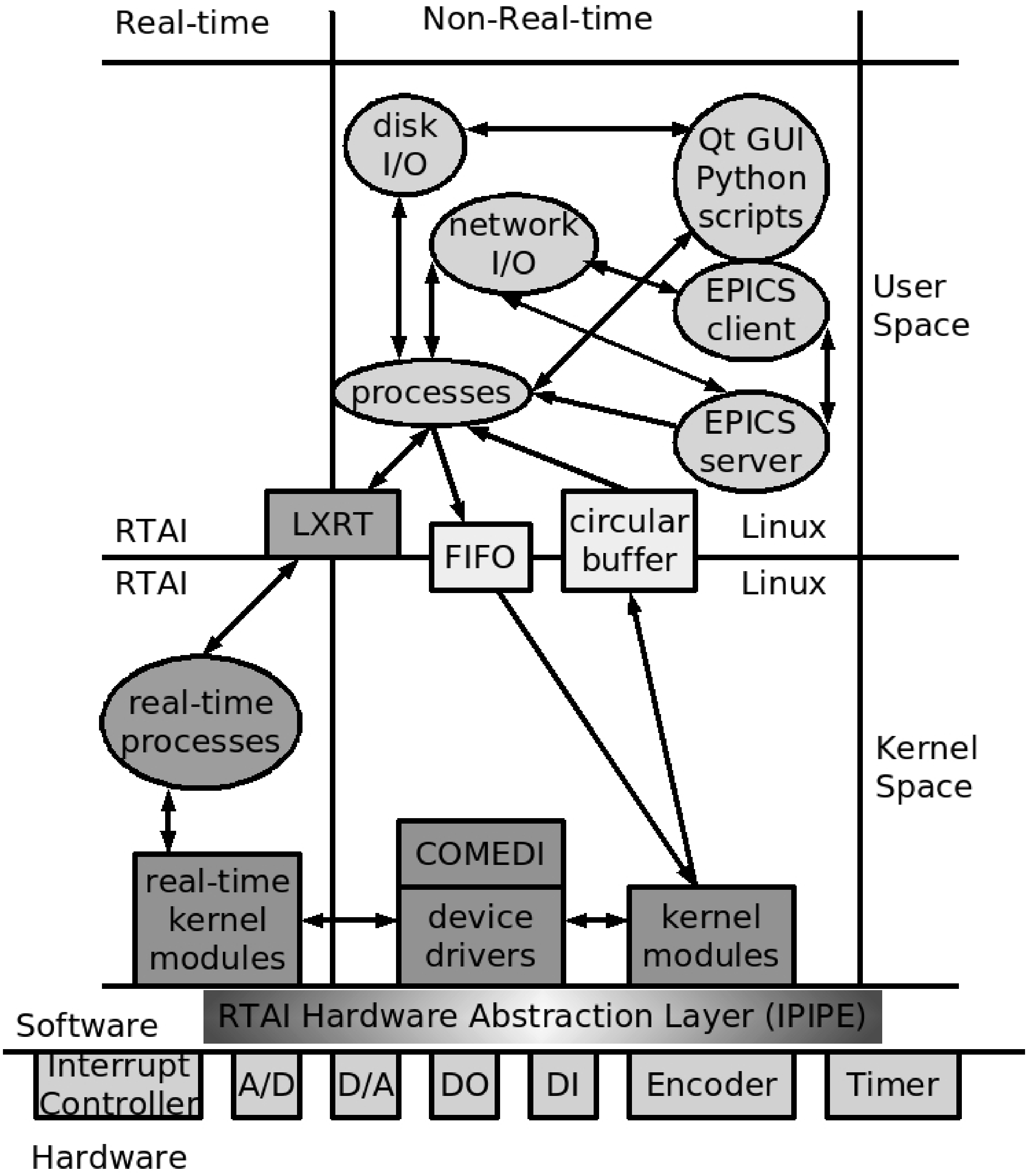

Fig 1. Software composition of the Graphical Real-time EPICS Controller (GREC) on top of data-acquisition hardware.

BNL-79366-2007-CP

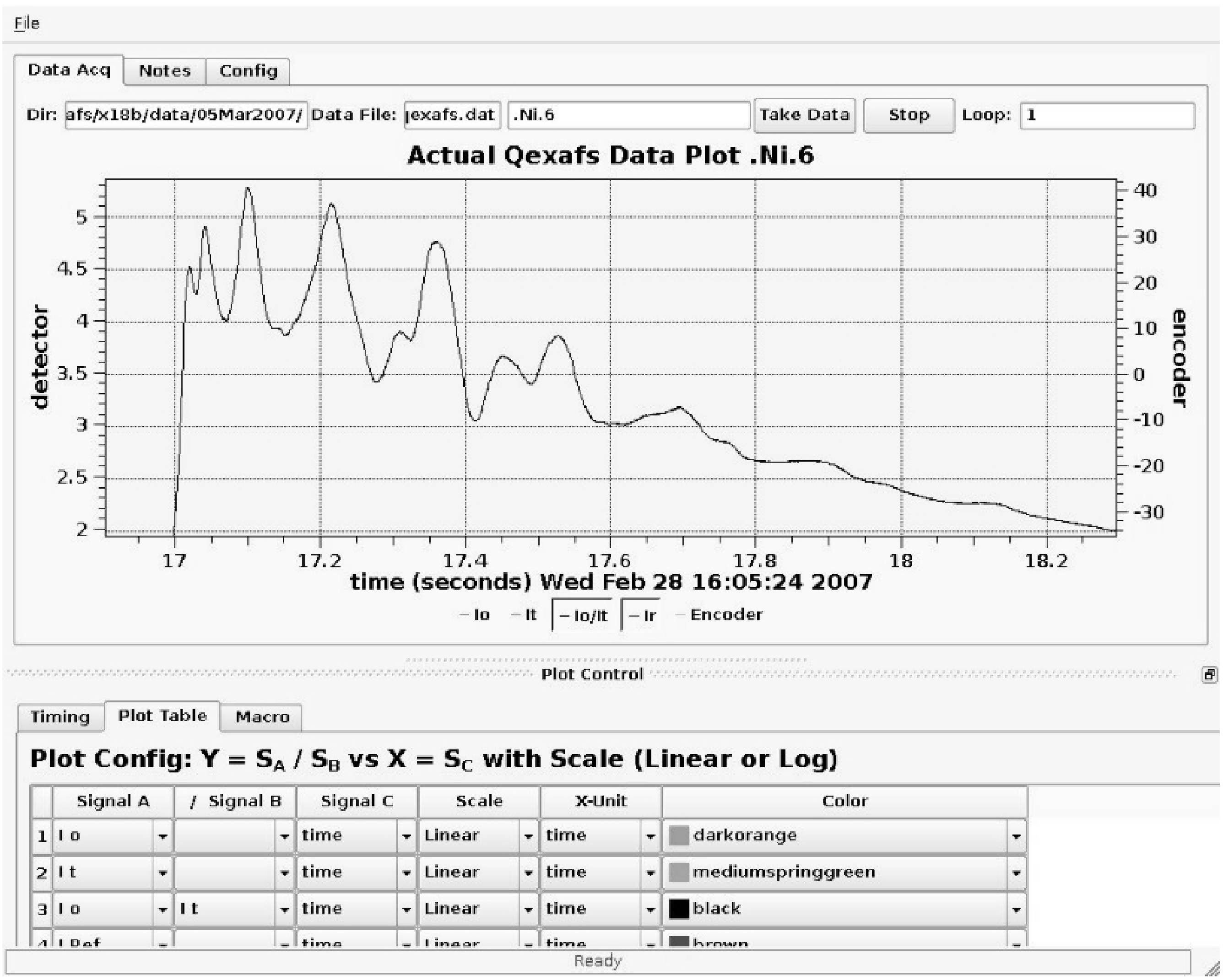

Fig 2. A zoomed 1.4 second plot of “Io / It” from a 30 second scan of Nickel-foil EXAFS at about 1/5 cycle/sec monochromator scanning frequency.